Post-Quantum Certificates for Electronic Travel Documents

Gaëtan Pradel[1,2], Chris J. Mitchell[2]

[1]INCERT, Luxembourg
gpradel@incert.lu
[2]Information Security Group, Royal Holloway, University of London, UK
me@chrismitchell.net

May 31, 2019

Abstract

Public key cryptosystems play a crucial role in the security of widely used communication protocols and in the protection of data. However, the foreseen emergence of quantum computers will break the security of most of the asymmetric cryptographic techniques used today, including those used to verify the authenticity of electronic travel documents. The security of international borders would thus be jeopardised in a quantum scenario. To overcome the threat to current asymmetric cryptography, post-quantum cryptography aims to provide practical mechanisms which are resilient to attacks using quantum computers. In this paper, we investigate the practicality of employing post-quantum digital signatures to ensure the authenticity of an electronic travel document. We created a special-purpose public key infrastructure based on these techniques, and give performance results for both creation and verification of certificates. This is the first important step towards specifying the next generation of electronic travel documents, as well as providing a valuable use case test for post-quantum techniques.

Keywords: Post-Quantum Cryptography, Certificates, Electronic Travel Document, PKI.

1 Introduction

Like many modern systems, the security of electronic passports and other electronic travel documents relies on public key cryptography. Whilst there are a number of very well-accepted and widely used public key schemes, the advent of large-scale, general-purpose, quantum computing will radically change the situation.

Quantum computers are built upon quantum mechanical phenomena, and can solve mathematical problems that classical computers cannot. Over the past few years, much effort has been devoted to building such a device, although

experts in the field suggest that it will be one or two decades before large scale quantum computers are a reality[1] [6, 18]. In the post-quantum era, the currently used asymmetric cryptographic techniques, i.e. integer factorization-based schemes (such as `RSA` [21]) and discrete logarithm-based schemes (such as Diffie-Hellman [7]), will be susceptible to being broken [20, 22]. This threatens the security of a wide range of systems, including the authenticity of electronic travel documents (the main focus of this paper).

In order to address this issue, as summarised by Bernstein and Lange [3], much recent effort has been devoted to developing post-quantum cryptographic schemes, i.e. schemes secure against attack using both quantum and classical computers. In parallel with this research effort, a number of major standardisation bodies have inaugurated projects to develop standards for post-quantum algorithms. Perhaps the most important of these is the competition led by the *National Institute of Standards and Technology (NIST)* [6]. So far, from an initial 82 submissions, after Round 2 of this competition only 26 schemes remain in the running for adoption[2].

Besides providing a portofolio of cryptographic algorithms resilient to quantum computers, this process of standardisation also aims to ensure that they are practical and can interoperate with current applications and protocols based on asymmetric cryptography. For example, Kampanakis et al. [15] showed that post-quantum X.509 certificates are viable for TLS-like communication protocols for use in a "post-quantum Internet". X.509 certificates are also commonly used to protect the authenticity and integrity of data inside electronic travel documents, namely the owner's data.

The focus of this paper is on a practical trial designed to test the feasibility of using currently available post-quantum cryptographic techniques in electronic travel documents. We have implemented a post-quantum *Public Key Infrastructure (PKI)* for electronic travel documents, and have also obtained results on its performance. Since this PKI is fundamental to the operation of security for electronic travel documents, the work described here can be seen as both preliminary research for the next generation of travel documents and also a testbed for evaluating post-quantum cryptographic techniques.

In Section 2 we describe how security is implemented for electronic travel documents. We then explain the development of the prototype post-quantum PKI in Section 3 and present the challenges we encountered in Section 4. Finally, we discuss our results in Section 5 and draw conclusions in Section 6.

2 Security for electronic travel documents

2.1 Electronic travel documents

For the last couple of decades, digital signatures have been used to protect electronic travel and national identity documents. The International Civil Aviation Organization (ICAO) started work on *Machine Readable Travel Documents (MRTDs)* as long ago as the late 1960s [9]. More recently, in 1998, work com-

[1] http://web.archive.org/web/20180817095418/http://www.research.ibm.com/5-in-5/quantum-computing/

[2] The results of Round 2 of the competition were published in January 2019, `https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions`

menced on *electronic MRTDs (e-MRTDs)*, resulting in a set of specifications covering the issue and border verification of such documents [9].

The specifications include protocols and mechanisms designed to protect the data inside the contactless chips embedded in the documents and allow border controllers to securely authenticate an issued e-MRTD. In order to verify an e-MRTD, the *inspection system (IS)*, used by border controllers for validating the authenticity of an e-MRTD, must perform the following steps:

1. access the contactless chip (see §2.4), where the IS proves to the chip that it is authorised to access it;
2. authenticate the card data (see §2.5), where the IS verifies that the data inside the chip (including the information in the data page[3]) is digitally signed by an authorised authority;
3. authenticate the contactless chip (see §2.6), where the chip proves to the IS that it is a genuine chip belonging to a genuine e-MRTD; and
4. (optionally) perform extended security protocols, e.g. to gain access to specific biometric data such as fingerprint or iris information.

2.2 Public Key Infrastructures

The security of e-MRTDs rest on an underlying PKI, the operation of which is the main focus of this paper. For our purposes a PKI (see, for example, Barak [2]) is a means of distributing trusted copies of public keys for asymmetric cryptographic techniques, and relies on the use of digital signatures. It involves a collection of *public key certificates*, digitally signed by *Certification Authorities (CAs)*, where each certificate contains a public key and associated information including the name of the owner, who is assumed to have the private key corresponding to the public key in the certificate.

The entities participating in the CA can be arranged as the vertices in a directed graph, where an edge goes from A to B if the certificate for B (Cert_B) was signed using A's private signature key, i.e. so that the public key of A can be used to verify Cert_B. Typically, a PKI will be arranged hierarchically, so that there is always a direct path (a *certificate chain*) from the *Root CA* to every *end-entity* (see Figure 1).

That is, if an entity has a trusted copy of the Root CA public key (typically distributed as a self-signed *Root CA certificate*), then a trusted copy of every end-entity's public key can be obtained in the following way. First construct a certificate chain from the Root CA to the end-entity, and then verify all the certificates in the chain in turn, at each stage verifying a certificate using the public key obtained by verifying the previous certificate.

[3]The document data page is the page containing the personal information of the document owner, such as his photo, name, date of birth and etc.

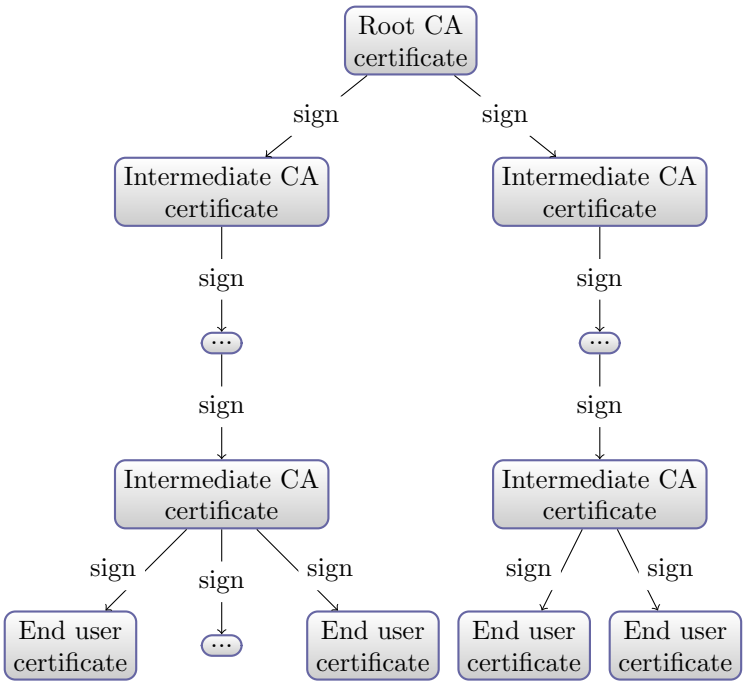

Figure 1: Typical PKI architecture

2.3 PKI for electronic travel documents

The PKI for e-MRTDs, e.g. e-passports, typically has three levels. The Root CAs are known as *Country Signing Certification Authorities (CSCAs)*, and, as the name suggests, are typically operated on behalf of a government department such as the Ministry of Foreign Affairs. Each country will operate a Root CSCA, and each such Root CSCA will have a digital signature key pair and a (self-signed) certificate for its public key, i.e. a public key certificate signed using the corresponding private key. These Root CSCAs are securely stored, and one of their role is to use their private signing keys to sign *Document Signer Certificates (DSCs)*, containing public keys of e-MRTD manufacturers. The corresponding private signature keys are used by the manufacturers to sign information held inside an e-MRTD.

In order to prove the authenticity and integrity of an e-MRTD at a border control, the self-signed root CSCA certificates are shared among states by bi-lateral exchanges, through states' *Master Lists*[4] or soon using the ICAO *Public Key Directory*[5].

A typical PKI for e-MRTDs is structured as shown in Figure 2.

[4]For example the German Master List: `https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/ElekAusweise/CSCA/GermanMasterList.html`.

[5]See `https://pkddownloadsg.icao.int/`.

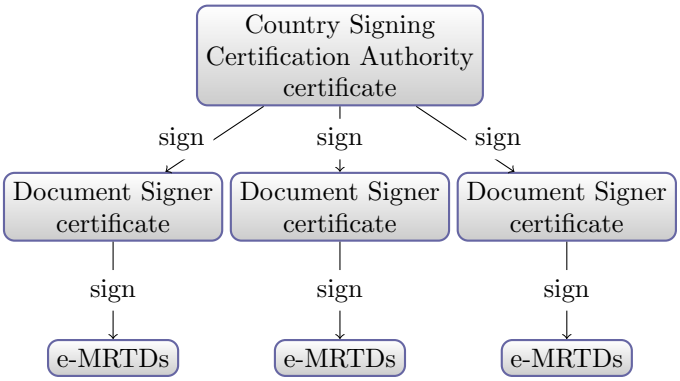

Figure 2: Typical PKI architecture for e-MRTDs

2.4 Access to the contactless chip

The first step for an IS is to gain authorised access to the e-MRTD's chip. It proves to the chip that it has the necessary authorisation using one of the following two protocols. To perform either of the following protocols, the IS shall have access to the *Machine Readable Zone (MRZ)* and be equipped to acquire it from the data page.

- *Basic Access Control (BAC)* is based on symmetric cryptography, and consists of a three-pass challenge-response protocol in accordance with Key Establishment Mechanism 6 of ISO/IEC 11770-2 [14] using two-key Triple-DES (see ISO/IEC 18033-3 [12]). A Message Authentication Code (MAC) is appended to the ciphertexts, computed using MAC algorithm 6 of ISO/IEC 9797-1 [13]. BAC may be deprecated in the future and is not authorised for use in e-MRTDs issued from January 1, 2018 [10].

- *Password Authenticated Connexion Establishment (PACE)* is based on asymmetric cryptography, and consists of a password-authenticated Diffie-Hellman key agreement protocol (see [5, 17]) which supplements and enhances BAC. The chip verifies that the IS is authorized to access its data and a secure communications channel is established.

2.5 Authentication of the data

In this paper we focus on this step, i.e. authentication of the chip-resident data, as verifying the validity of the e-MRTD data is probably the most important security function. This step includes only one protocol called *Passive Authentication (PA)*, so called because it does not require any computational capabilities (such as cryptographic operations) from the chip.

Data held in the contactless chip of an e-MRTD is stored in *Data Groups (DGs)*. Hashes of DGs are contained in the *Document Security Object (SO_D)*, which is used by the IS to verify the integrity and the authenticity of the data within the chip. The SO_D is digitally signed with the private key of a manufacturer, for which the corresponding public key is in a DSC signed with the private key associated to a root CSCA certificate (belonging to the government

agency on whose behalf the manufacturer is acting). The DSC must be placed in the SO_D so that the IS can retrieve it and use it to help verify the digital signature.

The PKI described in §2.3 is used in the following way to support data authentication. The IS retrieves the signed data and the DSC from the chip. The IS determines which CSCA (namely its associated private key) has been used to sign the DSC, and constructs a certificate chain from it. Verifying this chain (using the appropriate stored trusted Root CSCA public key) enables the appropriate DSC public key to be authenticated. This public key can finally be used to verify the signature on the chip data.

2.6 Authentication of the contactless chip

The third step for the IS is to authenticate the contactless chip, although this is not mandatory. This step enables the IS to verify that the chip is genuine, preventing copying and/or substitution, using one of the following three protocols.

- *Active Authentication* is based on asymmetric cryptography and requires the chip to sign a challenge sent by the IS with a private key sk held by the chip. This means that the chip has computational power, as it must perform a digital signature. The associated public key pk is accessible by the IS, and its authenticity has been already verified during Passive Authentication (see §2.5). After verification of the signed challenge, the IS is assured of the authenticity of the chip. This technique raises a privacy issue, as each generated signature could be logged. The owner of an e-MRTD (and thus the owner of the private key used to sign the challenges) could then be traced using the logged signatures. The Chip Authentication protocol (see below) has been devised in order to mitigate this risk.

- *Chip Authentication* is based on asymmetric cryptography, more precisely on a variant of the Diffie-Hellman key agreement protocol. In addition to guaranteeing the authenticity of the chip, it also provides authentication of the data inside the chip and a secure communication channel between chip and IS. Moreover, as the exchanged keys are ephemeral, it prevents any tracing of the e-MRTD's owner. The static key pair used for the protocol is stored inside the chip, where the private key is in secure memory and the public key is accessible to the IS.

- *PACE with Chip Authentication Mapping* is a combination of PACE (§2.4) and Chip Authentication (§2.6), optimised for performance.

3 Building a post-quantum PKI for electronic travel documents

As discussed in §2.5, the authenticity of e-MRTD chip data is verified using the PA protocol. This protocol relies on the PKI established by states through their networks of CSCAs. Thus, to ensure that PA continues to provide security in the post-quantum world, a *post-quantum PKI (pqPKI)*, i.e. a PKI based on the

architecture presented in §2.3 but using post-quantum cryptography, is needed. To verify the practicality of building and operating such a PKI, we have built a proof-of-concept implementation which we next describe.

3.1 Design

For the purposes of this proof-of-concept, the PKI architecture for e-MRTDs presented in §2.3 can be simplified without any loss of generality. The proof-of-concept PKI is thus composed of one CSCA certificate and one DSC, as shown in Figure 3.

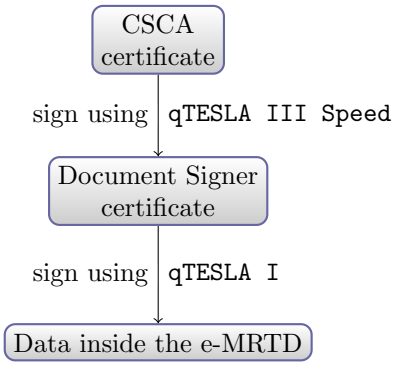

Figure 3: Proof-of-concept post-quantum PKI architecture

Both types of certificate follow the standard structure for an X.509 certificate, signed using a post-quantum digital signature algorithm, e.g. as presented in [15], although the certificates must also be compliant with ICAO Doc 9303 Part 12 [11]. These specifications define the extensions and the associated values for each type of certificate in the e-MRTD PKI, with the details depending on their role in this structure, i.e. their *certificate profile*.

The CSCA certificate is self-signed and the associated private key is used to sign the private key associated with the DSC. The DSC is then normally used to sign an e-MRTD document, in our case this involves signing data of any type, ideally a SO_D (see Section 2).

The prototype targeted the replacement of any cryptographic algorithm which could be potentially vulnerable to cryptanalysis using quantum computers. Two types of cryptographic primitive are used in a digital signature algorithm:

1. a collision-resistant hash function; and
2. a public key encryption algorithm.

Quantum computers pose a much greater threat to the latter [3], but both primitives must be quantum-resistant to yield a secure digital signature algorithm. We focused on lattice-based digital signature algorithms such as `qTESLA` [1] and `CRYSTALS-Dilithium` [8], although in this paper we present results only

for qTESLA. We chose to use qTESLA[6] with SHA-3 hash function [4],the former for its performances and its claimed security and the latter is believed to be secure (for preimage security) in a post-quantum world [3]. The qTESLA family includes different parameter sets corresponding to the different NIST's security categories[7]. We used qTESLA I and qTESLA III Speed corresponding respectively to NIST's security category 1 and 3.

To build a post-quantum system, we accomplished the following steps in chronological order:

1. we generated a highly secure key pair to be associated with the root CSCA certificate;

2. we generated a *Certificate Signing Request (CSR)* with the previously generated key pair and the CSCA certificate profile that we then self-sign, thus this would be our CSCA certificate;

3. we generated a secure key pair, usually considered as "less secure" but with a shorter lifespan than the associated key pair of the root CSCA certificate because of its shorter length, although this choice results in better computation performances for the digital signature algorithm and key generation;

4. we generated another CSR with the previously generated key pair and the DSC certificate profile that we sign with the previously generated CSCA certificate associated private key, generating thus a DSC;

5. we sign some hashed random data with the private key associated with the DSC to complete the chain shown in Figure 3.

3.2 Implementation

To implement our architecture, we used a fork of OpenSSL combined with the library liboqs from the *Open Quantum Safe (OQS)* project [23]. OpenSSL is an open-source implementation of the *Transport Layer Security (TLS)* and *Secure Sockets Layer (SSL)* protocols, and incorporates a widely used cryptographic primitives library. It was not designed to establish PKIs, such as a PKI for e-MRTDs; however, despite this we decided to use this software because of its wide use and flexibility. Also, OpenSSL is implemented in C, as are all the submissions to the NIST Post-Quantum Standardization project[8], enabling straightforward integration.

liboqs is a open-source library in C of post-quantum algorithms, which has been integrated into prototype forks of OpenSSL or OpenSSH. liboqs includes algorithms from the NIST Post Quantum Standardization Project. To generate the PKI for e-MRTDs described in §3.1, we implemented an OpenSSL configuration file that caused it to issue certificates with the appropriate extensions. The configuration file included all the certificate components and extensions

[6] At the time of the execution of the work, qTESLA was not considered as insecure and presented good performances although some latest comments argue with the security statement: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-2/official-comments/qTESLA-round2-official-comment.pdf

[7] https://csrc.nist.gov/publications/detail/fips/199/final

[8] https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions

needed by each certificate type, as defined in the relevant certificate profile, i.e. a CSCA certificate or a DSC, as specified in ICAO Doc 9303 Part 12 [11].

4 Challenges

OpenSSL is an implementation of SSL/TLS, and is not designed to generate and manage a PKI producing certificates for signing electronic travel documents according to ICAO Doc 9303 Part 11 and Part 12 [10, 11]. For example, extensions such as *Private Key Usage period*, which are required by ICAO, cannot be set up with OpenSSL, although they can be displayed. To overcome this difficulty, we took advantage of the fact that OpenSSL allows integration of ad hoc extensions created by the user via the Arbitrary Extension module[9]. This allows an implementer to encode arbitrary extensions in created certificates[10].

A problem was encountered when trying to create a certificate chain. Although the software produced chains using well-established digital signature schemes, it refused to produce them for the chosen post-quantum algorithms. We reported the problem to the authors of the `liboqs` library, and simultaneously worked on a resolution. The issue has now been resolved and the documentation for the software has been updated[11].

5 Results

We generated certificates according to the two certificate profiles described in Section 3 (CSCA certificate and DSC) for three algorithms and two key lengths, and in each case measured the memory size and generation time. To perform the operations we used an Ubuntu 18.04.2 LTS $x86\_64$ virtual machine with 2GB of RAM and one core Intel(R) Xeon(R) Silver 4110 CPU 2.10GHz. Two of the three algorithms used were current signature schemes (RSA and ECDSA [16] with the Brainpool parameters [19]), which were included for comparison purposes. Note that in these experiments certificate generation included both key pair generation and signing of the certificate, apart from Figure 4, Figure 5 and Figure 6 for which the different steps of the certificate issuance process have been clearly separated. Table 1 summarizes the algorithms and key lengths used for the two certificate types.

	CSCA certificate	DSC
`qTESLA`	`qTESLA III Speed` with `SHAKE-256`	`qTESLA I` with `SHAKE-128`
`RSA`	4096 bits with `SHA-256`	2048 bits with `SHA-256`
`Brainpool`	384 bits with `SHA-256`	224 bits with `SHA-256`

Table 1: Algorithms and key lengths by certificate type

To construct a post-quantum PKI, we separated certificate generation into three steps, according to the process described in §3.1, as follows:

[9] `https://www.openssl.org/docs/manmaster/man5/x509v3_config.html`

[10] An example of such an ad hoc extension is given at: `http://openssl.6102.n7.nabble.com/Private-Key-Usage-Period-td28401.html`

[11] See resolution in `https://github.com/open-quantum-safe/openssl/issues/68`

1. generation of the key pair;
2. generation of the CSR; and
3. generation of the certificate (including the signature of the previously generated CSR).

To be consistent with the associated certificate profile, the CSCA certificates were all self-signed and the DSCs were signed with a CSCA private key from the same algorithm family, i.e. a DSC including a `qTESLA I` public key was signed with a `qTESLA III Speed` private key. Each of the three steps corresponds to an OpenSSL command and we measured the execution time for 1000 iterations. The results are shown in Figure 4, Figure 5 and Figure 6 respectively for each generation step. `qTESLA` and `Brainpool` have similar performance results in each of the presented cases, although `RSA` demonstrates much slower key pair generation and slightly slower CSR and certificate generation.

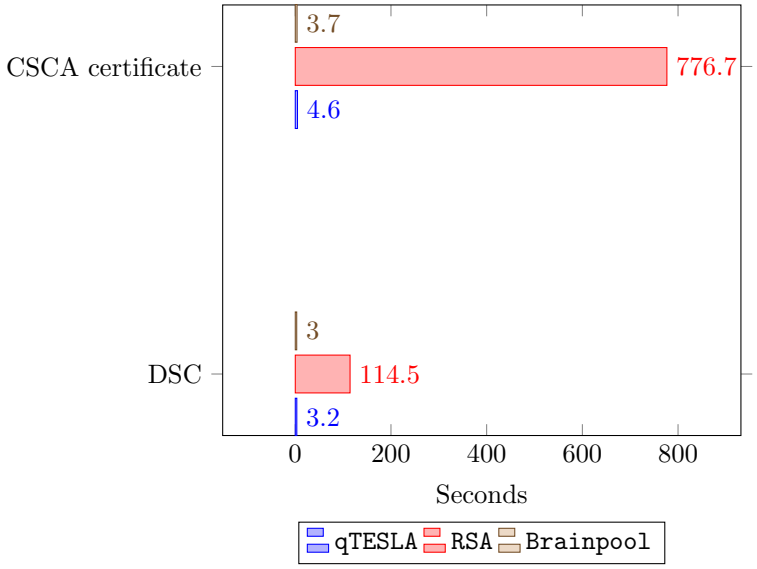

Figure 4: Time in seconds(s) to generate 1000 key pairs

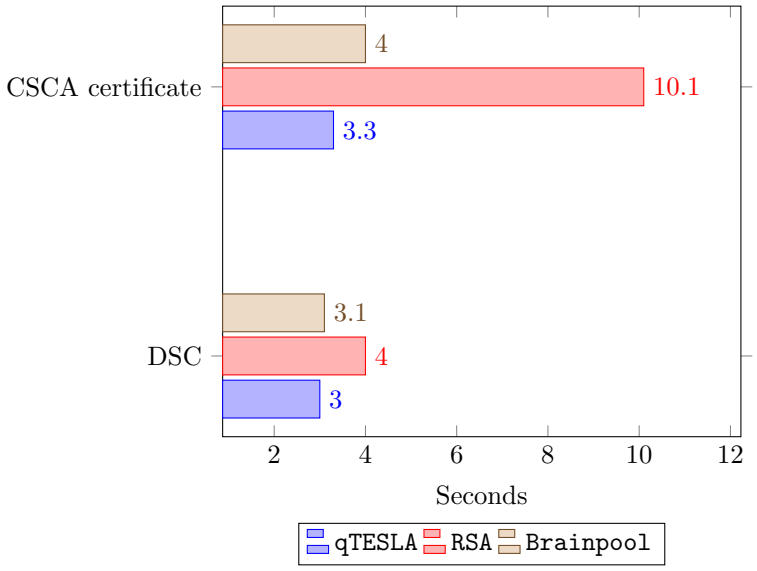

Figure 5: Time in seconds(s) to generate 1000 CSRs

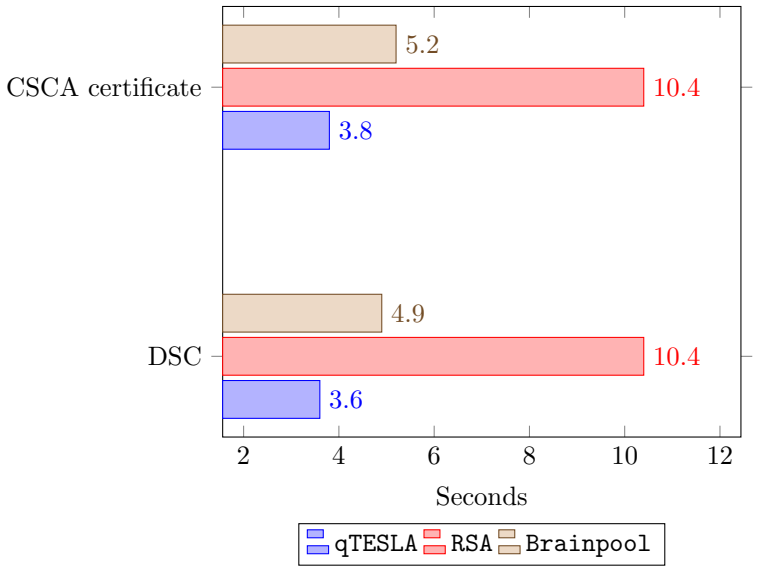

Figure 6: Time in seconds(s) to generate 1000 certificates

In addition, we generated as many certificates as possible during a 5 second period for each certificate profile, algorithm and key length. The generation throughput for the post-quantum algorithms is actually greater than for the two classical algorithms. In particular for `RSA`, as the generation of the key pair is not efficient, we managed to generate on average only a few CSCA certificates.

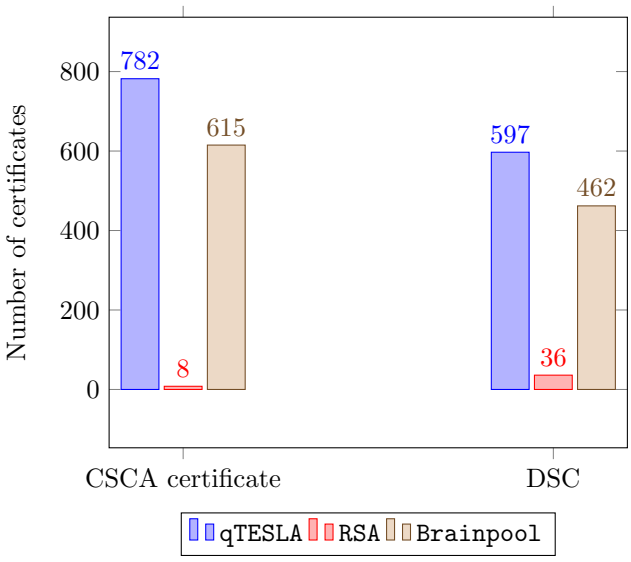

Figure 7: Throughput of certificates in a 5 seconds period

We then generated 1000 minimal PKIs for e-MRTDs, i.e. a self-signed CSCA certificate and a DSC signed by this CA. The results for `qTESLA` and `Brainpool` are quite similar even though `qTESLA` is significantly faster, but using `RSA` is clearly less efficient as it take at least 50 times longer (see Figure 8) due to its poor performance for key pair generation (see Figure 4).

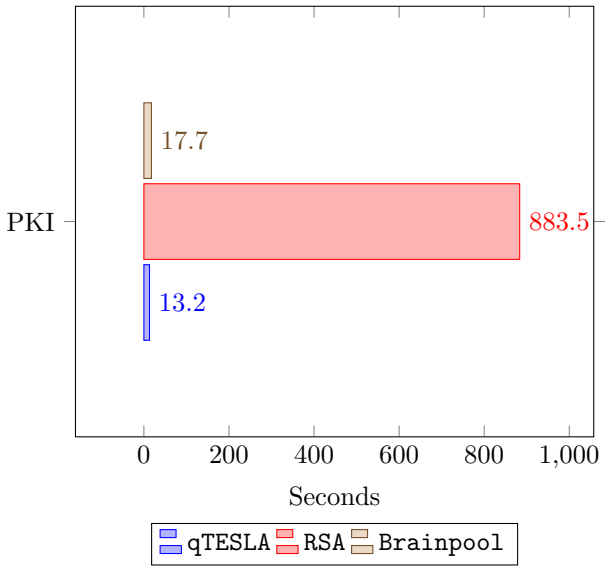

Figure 8: Time in seconds(s) to generate 1000 minimal PKIs

To this point, the results were clearly in favour of the post-quantum alternative. However, we also looked into the memory space necessary to store the various certificates, and the certificates based on the two classical algorithms

were significantly smaller (see Figure 9). Unsurprisingly, we obtain similar results for the generated key pairs (see Figure 10).

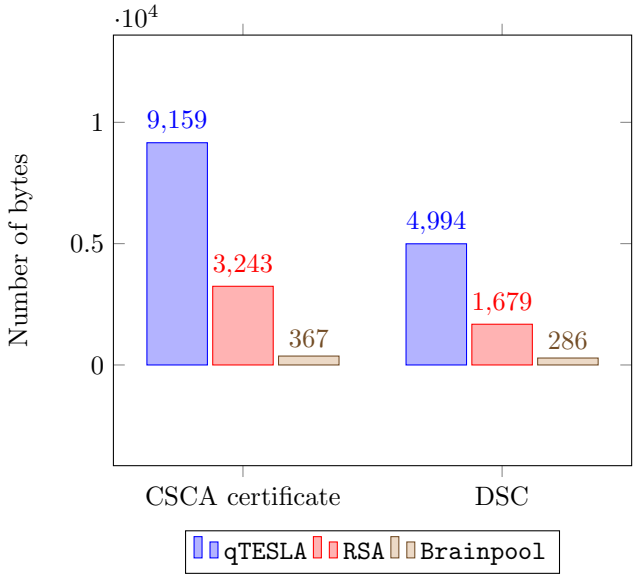

Figure 9: Size of certificates in bytes

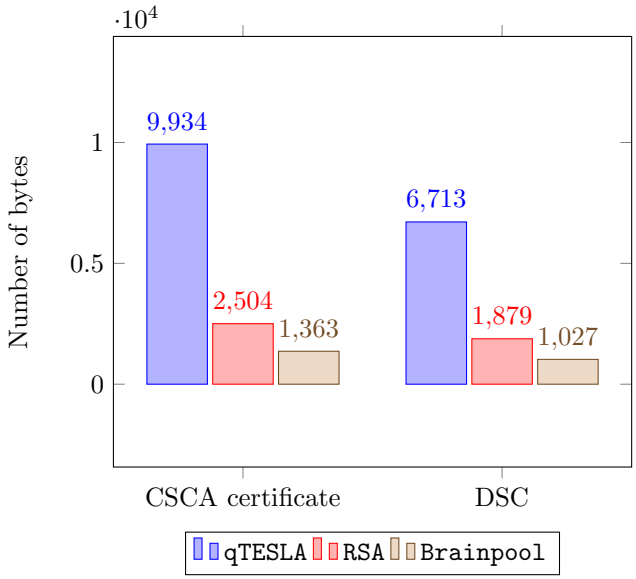

Figure 10: Size of key pairs in bytes

Over and above these somewhat abstract performance results, we wanted to consider how a switch to post-quantum algorithms would affect the "real world". That is, we wanted to assess the impact of a move to the post-quantum world on the generation and management of CSCA certificates and DSCs for

government authorities.

We use as an example Luxembourg, in which the management of the PKI for generating the digital signatures of electronic travel documents has been assigned to a public agency[12] under the authority of the Ministry of Economy and the Ministry of Foreign Affairs. Here we do not consider the capacities of the contactless chips inside the electronic travel documents.

Typically, the infrastructure of a PKI is based on servers and Hardware Security Modules (HSMs), and can be arbitrarily expanded. CSCA certificates are issued every 3 to 5 years for renewal, and thus the overload in term of performance and memory will not be an issue. The DSCs, have to be renewed under two criteria: their lifespan or the number of signatures they performed. As best practice, both of them shall be low, to avoid producing many digital signatures in line with one single key. Suppose that their lifespan is one month that they can sign at most 100 000 electronic travel documents. Luxembourg is around 600 000 inhabitants, but only half are Luxembourgers so we can assume that the production of electronic travel documents is quite light compare to other countries. With this number of inhabitants, each DSC does not reach the threshold of 100 000 digital signatures, but we can suppose that they do. Key generations are slower for our tested classical algorithms, and digital signatures are faster for the `qTESLA` family.

If we do not ignore the capacities of the contactless chip inside the electronic travel documents, we shall only be certain that we do have the memory space necessary to store the post-quantum certificates and signatures necessary to perform PA, which do not require any computational power from the chip (see §2.5). Current chips[13] for electronic travel documents can have memory space as much as 160 Kbytes in EEPROM and 280 KBytes in User ROM. Those sizes would be largely enough to store a post-quantum certificate (see Figure 9) and digital signatures based on `qTESLA` [1].

6 Conclusion and future work

As in the work of [15], the results of this paper showed that post-quantum X.509 certificates can be used in current applications such as that on which we focused: electronic travel documents. We used the cryptographic `qTESLA` family as an example for our proof-of-concept, and showed that, the performance for key generation and digital signature are better than some classical cryptographic asymmetric techniques (namely `RSA` and `Brainpool`). At the same time, whilst the memory requirements increase, the change is not sufficiently large to make the algorithms impractical. Of course, eMRTDs produced with a post-quantum algorithm such as `qTESLA` would not be compliant with ICAO Doc 9303 Part 12[11] which defines the algorithms to can be used. ICAO will have to update their specifications for the post-quantum era in order to ensure the security of the electronic travel documents. For this feasibility test of post-quantum PKI for electronic travel documents, we decided to use OpenSSL to have freedom and ease of use, but this tool is not optimized or even designed for such a specific PKI.

[12] `https://www.incert.lu`

[13] See for example these contactless security cryptocontroller: `https://www.infineon.com/cms/en/product/security-smart-card-solutions/security-controllers/sle-78/`.

Possible future work is to use JMRTD[14], an open source Java implementation for MRTD standards. This tool uses *The Legion of the Bouncy Castle*[15], a cryptographic techniques library which has included `qTESLA` since its last release[16].

The next generation of electronic travel documents will be based on post-quantum cryptographic techniques, but do not exist yet, as far as we are aware. This paper focuses only on one of the three steps verifying the authenticity of an electronic travel document, but the other two steps require also cryptographic asymmetric techniques that will need to be quantum-resistant.

Finally, governmental authorities managing a CSCA usually manage another type of CA, which is known as the *Country Verifying Certification Authority (CVCA)*. A CVCA is used to issue *Card Verifiable Certificates (CVCs)* to control authorities (such as the national police) so they can access to the fingerprints and the irises (if they are included) in the controlled electronic travel document.

7 Acknowledgements

The present project is supported by the National Research Fund, Luxembourg.

References

[1] Erdem Alkim, Paulo S. L. M. Barreto, Nina Bindel, Patrick Longa, and Jefferson E. Ricardini. The lattice-based digital signature scheme qtesla. Cryptology ePrint Archive, Report 2019/085, 2019. `https://eprint.iacr.org/2019/085`.

[2] Boaz Barak. The complexity of public-key cryptography. Cryptology ePrint Archive, Report 2017/365, 2017. `https://eprint.iacr.org/2017/365`.

[3] Daniel J. Bernstein and Tanja Lange. Post-quantum cryptography — dealing with the fallout of physics success. Cryptology ePrint Archive, Report 2017/314, 2017. `https://eprint.iacr.org/2017/314`.

[4] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 313–314. Springer, 2013.

[5] BSI. Elliptic curve cryptography. Technical guideline, Federal Office for Information Security, Bonn, Germany, 2018.

[6] Lily Chen, Stephen Jordan, Yi-Kay Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. Report on post-quantum cryptography. Report, US Department of Commerce, National Institute of Standards and Technology, 2016.

[14] `https://jmrtd.org/`

[15] `http://www.bouncycastle.org/`

[16] Update released in February 2019 after we completed this work: `http://www.bouncycastle.org/latest_releases.html`

[7] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Information Theory*, 22(6):644–654, 1976.

[8] Leo Ducas, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehle. CRYSTALS – Dilithium: Digital Signatures from Module Lattices. Cryptology ePrint Archive, Report 2017/633, 2017. `https://eprint.iacr.org/2017/633`.

[9] ICAO. Doc 9303 — Machine Readable Travel Documents — Part 1: Introduction. Technical report, International Civil Aviation Organization, Montréal, CA, 2015. Seventh Edition.

[10] ICAO. Doc 9303 — Machine Readable Travel Documents — Part 11: Security Mechanisms for MRTDs. Technical report, International Civil Aviation Organization, Montréal, CA, 2015. Seventh Edition.

[11] ICAO. Doc 9303 — Machine Readable Travel Documents — Part 12: Public Key Infrastructure for MRTDs. Technical report, International Civil Aviation Organization, Montréal, CA, 2015. Seventh Edition.

[12] ISO Central Secretary. Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers. Standard ISO/IEC 18033-3:2010, International Organization for Standardization, Geneva, CH, 2010.

[13] ISO Central Secretary. Information technology — Security techniques — Message Authentication Codes (MACs) — Part 1: Mechanisms using a block cipher. Standard ISO/IEC 9797-1:2011, International Organization for Standardization, Geneva, CH, 2011.

[14] ISO Central Secretary. IT Security techniques — Key management — Part 2: Mechanisms using symmetric techniques. Standard ISO/IEC 11770-2:2018, International Organization for Standardization, Geneva, CH, 2018.

[15] Panos Kampanakis, Peter Panburana, Ellie Daw, and Daniel Van Geest. The viability of post-quantum x.509 certificates. Cryptology ePrint Archive, Report 2018/063, 2018. `https://eprint.iacr.org/2018/063`.

[16] Cameron F. Kerry, Acting Secretary, and Charles Romine Director. FIPS PUB 186-4 FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION Digital Signature Standard (DSS), 2013.

[17] RSA Laboratories. Diffie-hellman key-agreement standard. Standard, RSA Security Inc., Redwood City, California, USA, 1993.

[18] Matteo Mariantoni. Building a superconducting quantum computer. Invited Talk PQCrypto 2014, 2017. `https://www.youtube.com/watch?v=wWHAs--HA1c`.

[19] Johannes Merkle and Manfred Lochter. Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation. RFC 5639, March 2010.

[20] John Proos and Christof Zalka. Shor's discrete logarithm quantum algorithm for elliptic curves. *Quantum Information & Computation*, 3(4):317–344, 2003.

[21] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[22] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[23] Douglas Stebila and Michele Mosca. Post-quantum key exchange for the internet and the open quantum safe project. In Roberto Avanzi and Howard M. Heys, editors, *Selected Areas in Cryptography — SAC 2016 — 23rd International Conference, St. John's, NL, Canada, August 10-12, 2016, Revised Selected Papers*, volume 10532 of *Lecture Notes in Computer Science*, pages 14–37. Springer, 2016.